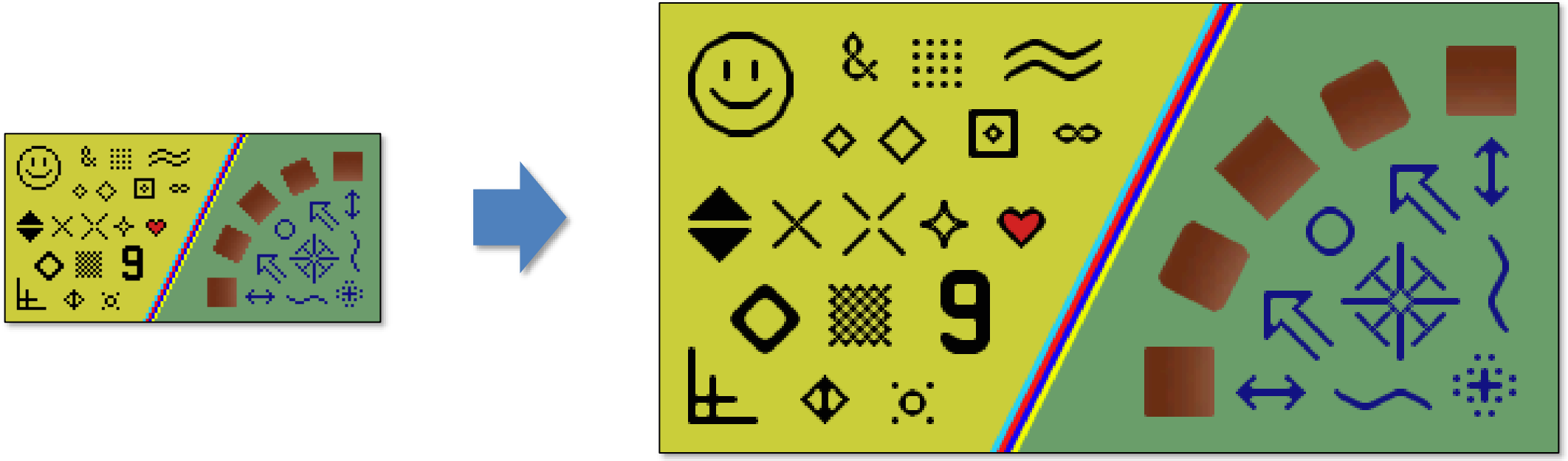
# CS5670: Computer Vision

## Image Resampling & Interpolation

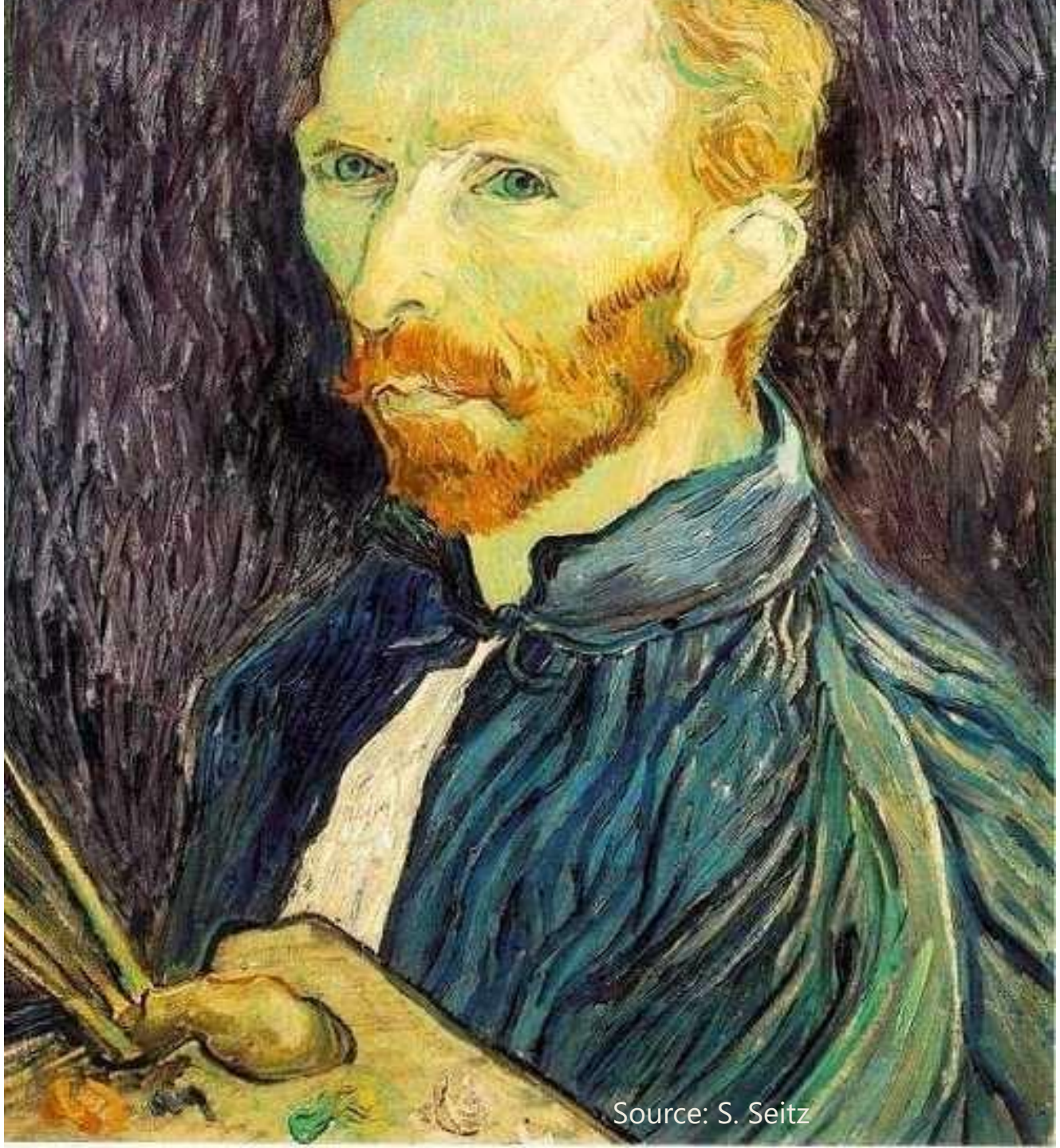


# Announcements

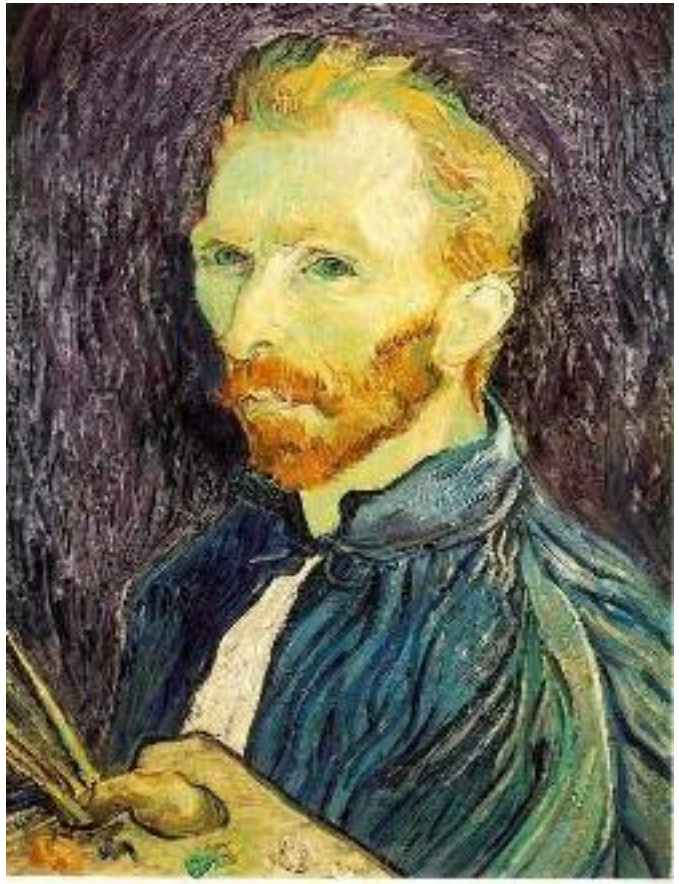
- Project 1 released, due Thursday, February 25 by 11:59pm
  - Project to be done solo (teams of one)
  - Artifact due Monday, March 1 by 11:59pm
- First quiz next week, TBA. Will accommodate other time zones.

# Image scaling

This image is too big to fit on the screen.  How can we generate a half-sized version?



# Image sub-sampling



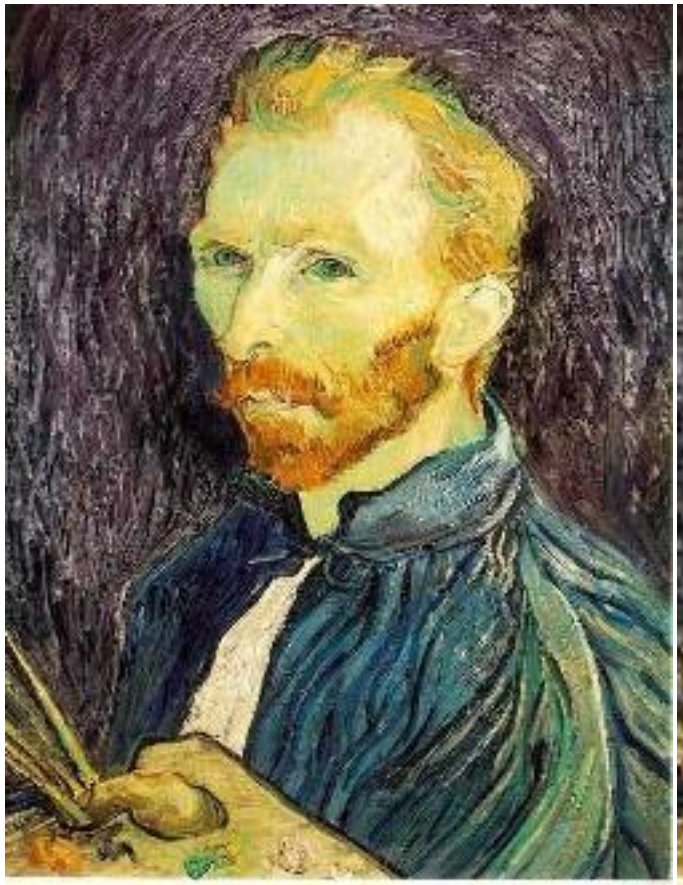
1/4

1/8

Throw away every other row and column to create a 1/2 size image
- called *image sub-sampling*

Source: S. Seitz

# Image sub-sampling



1/2

1/4 (2x zoom)

1/8 (4x zoom)

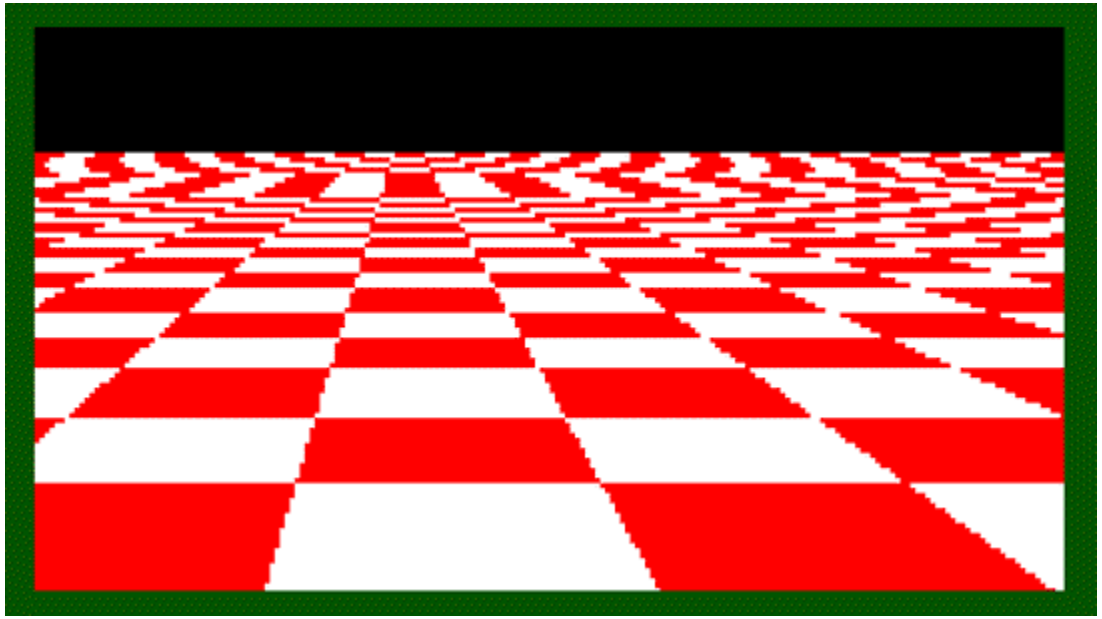
Why does this look so crufty?

Source: S. Seitz

# Image sub-sampling – another example

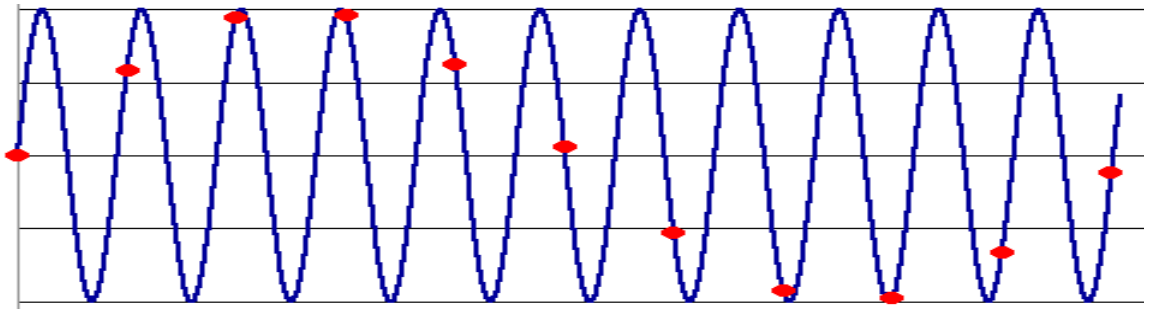
Source: F. Durand

# Even worse for synthetic images



Source: L.

# Aliasing



- Occurs when your sampling rate is not high enough to capture the amount of detail in your image
- Can give you the wrong signal/image—an *alias*

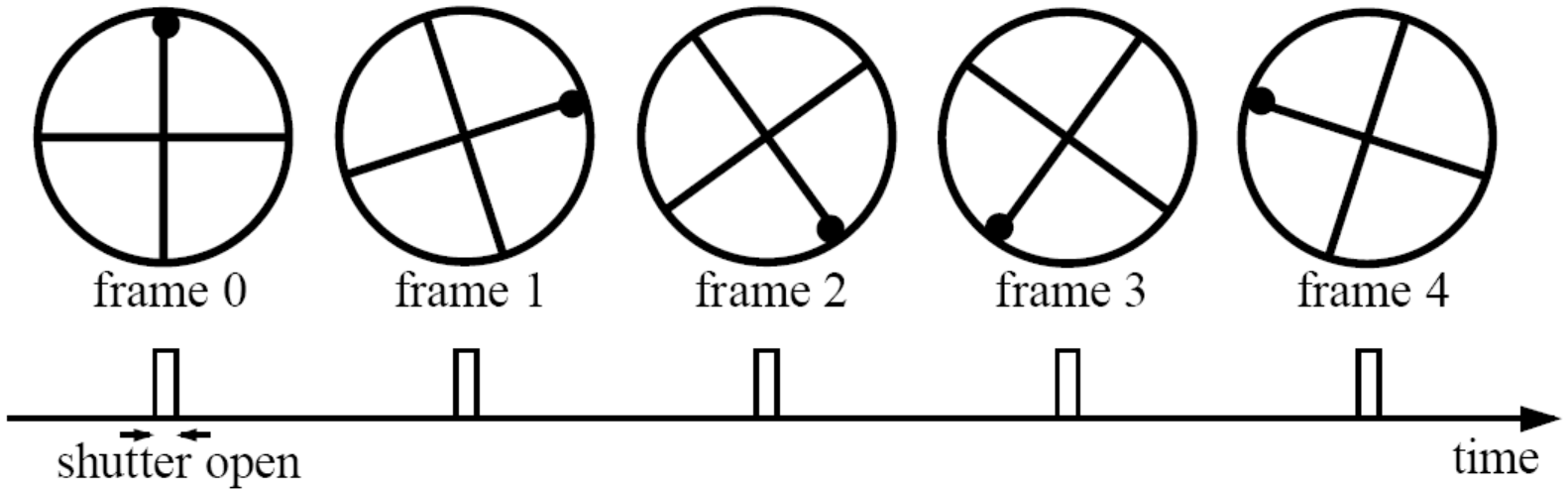
- To do sampling right, need to understand the structure of your signal/image
- Enter Monsieur Fourier…
  - "But what is the Fourier Transform? A visual introduction." https://www.youtube.com/watch?v=spUNpyF58BY
- To avoid aliasing:
  - sampling rate ≥ 2 * max frequency in the image
    - said another way: ≥ two samples per cycle
  - This minimum sampling rate is called the **Nyquist rate**

Source: L. Zhang

# Wagon-wheel effect

Imagine a spoked wheel moving to the right (rotating clockwise). Mark wheel with dot so we can see what's happening.

If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):




Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

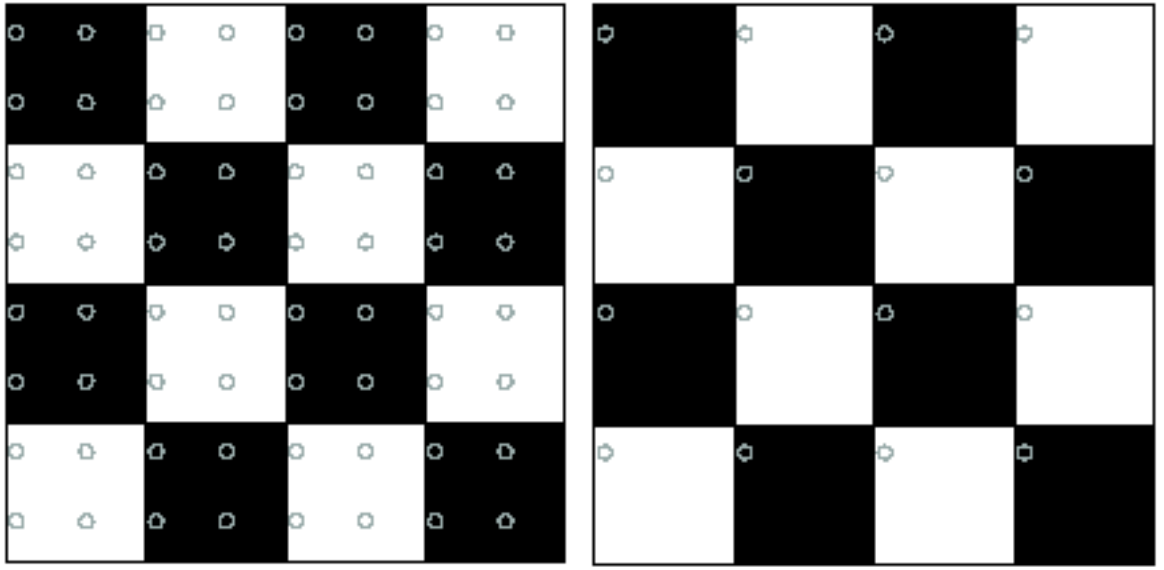
(See http://www.michaelbach.de/ot/mot-wagonWheel/index.html)

Source: L. Zhang

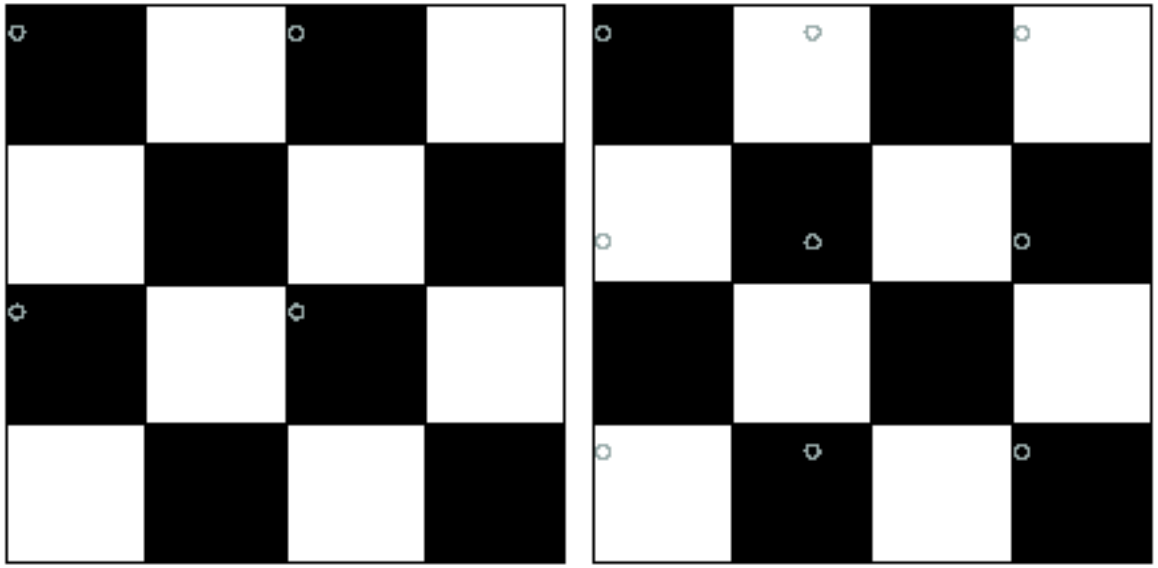
# Wagon-wheel effect

https://en.wikipedia.org/wiki/Wagon-wheel_effect

# Nyquist limit – 2D example



Good sampling



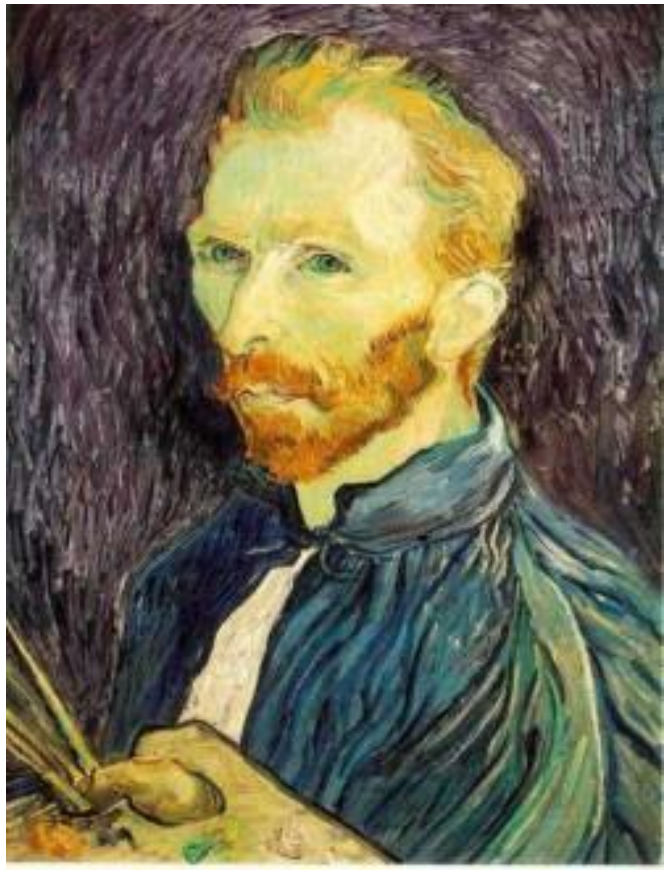
Bad sampling

# Aliasing

- When downsampling by a factor of two
  - Original image has frequencies that are too high

- How can we fix this?

# Gaussian pre-filtering



Gaussian 1/2

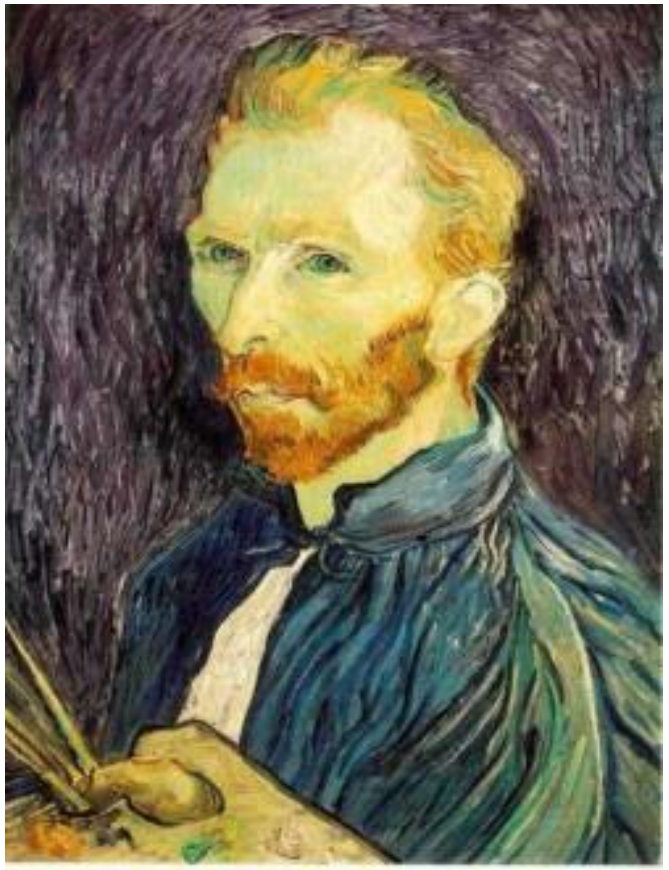
G 1/4

G 1/8

- Solution: filter the image, *then* subsample

Source: S. Seitz

# Subsampling with Gaussian pre-filtering



Gaussian 1/2

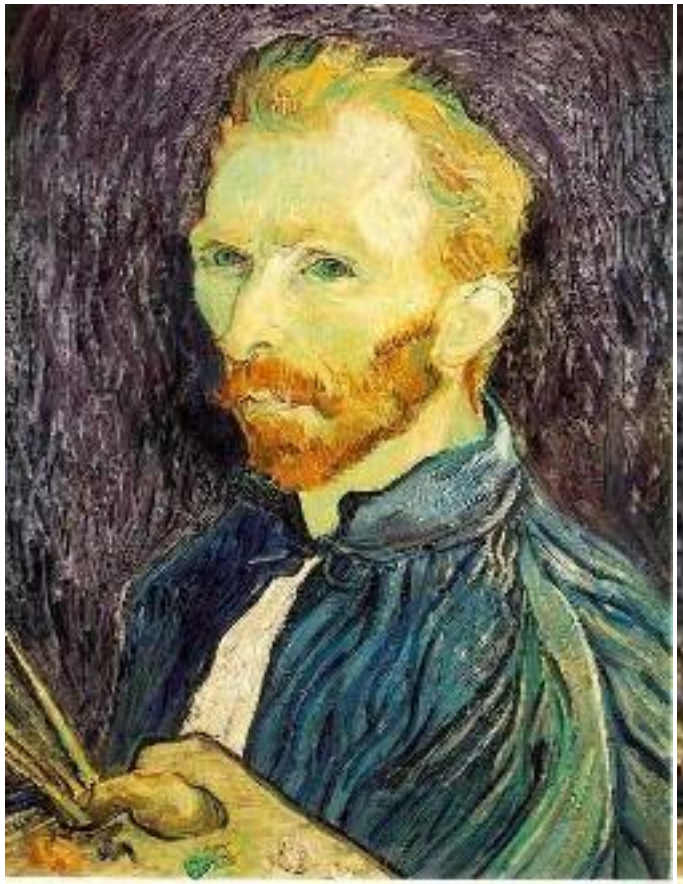
G 1/4

G 1/8

- Solution: filter the image, *then* subsample

Source: S. Seitz

# Compare with...



1/2

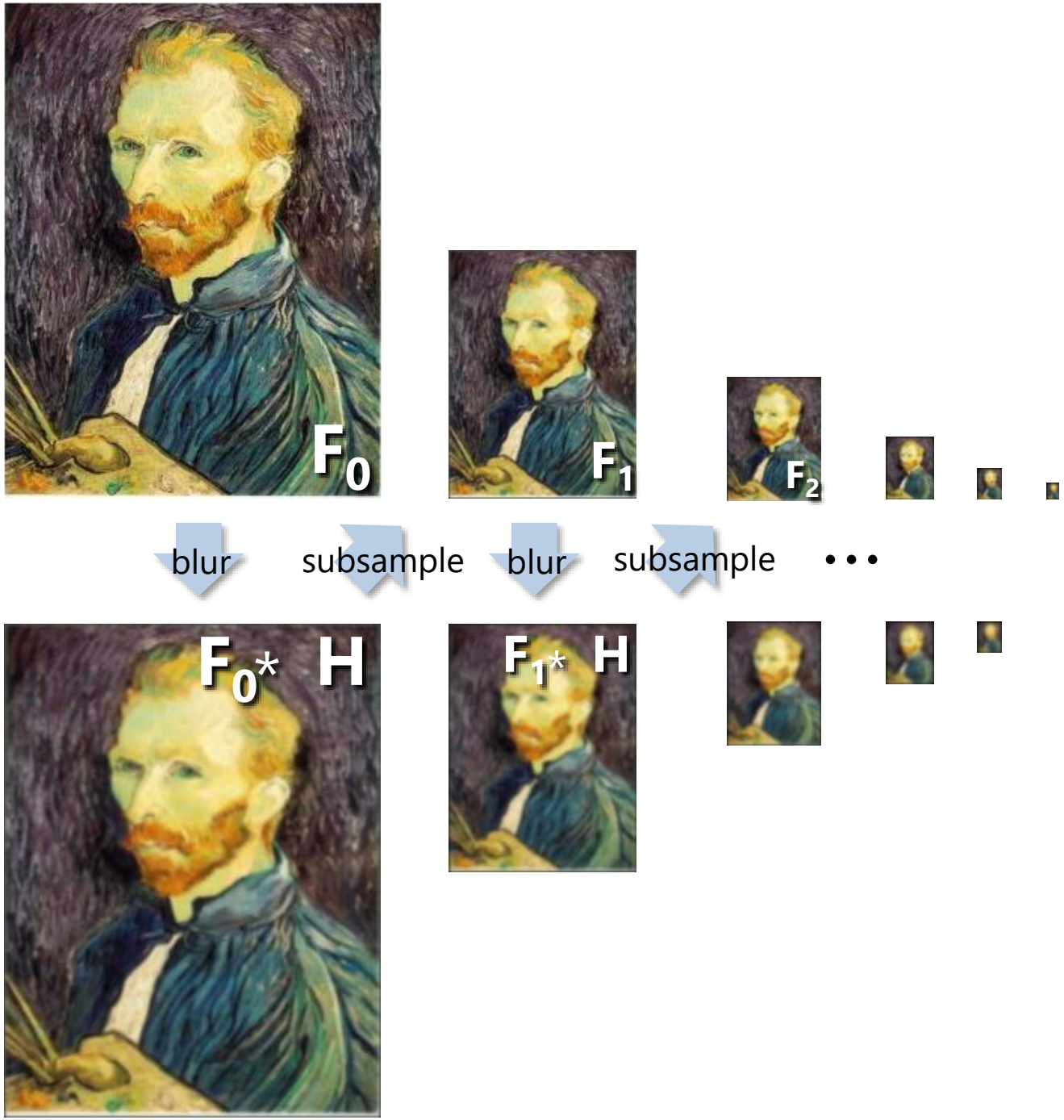
1/4 (2x zoom)

1/8 (4x zoom)

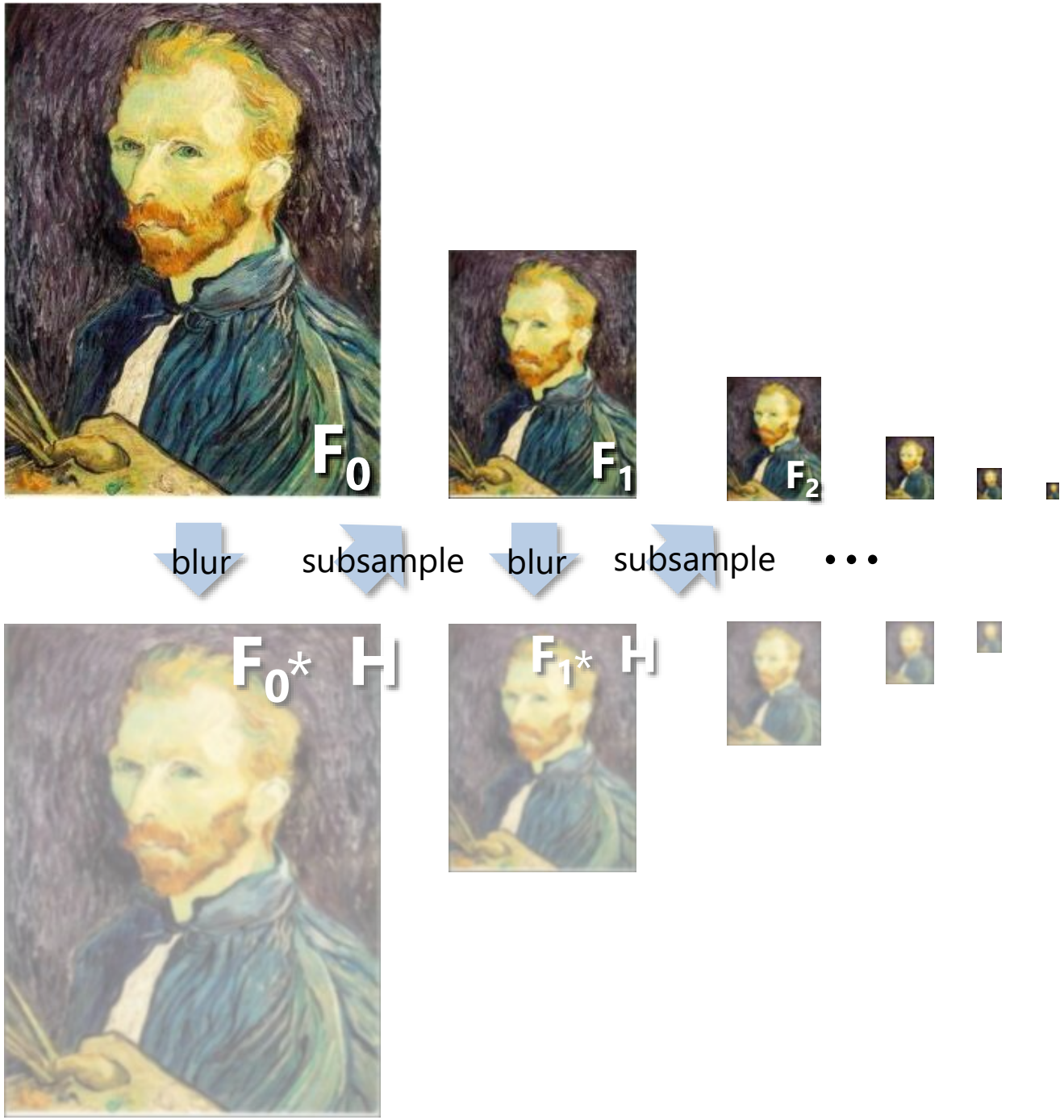
Source: S. Seitz

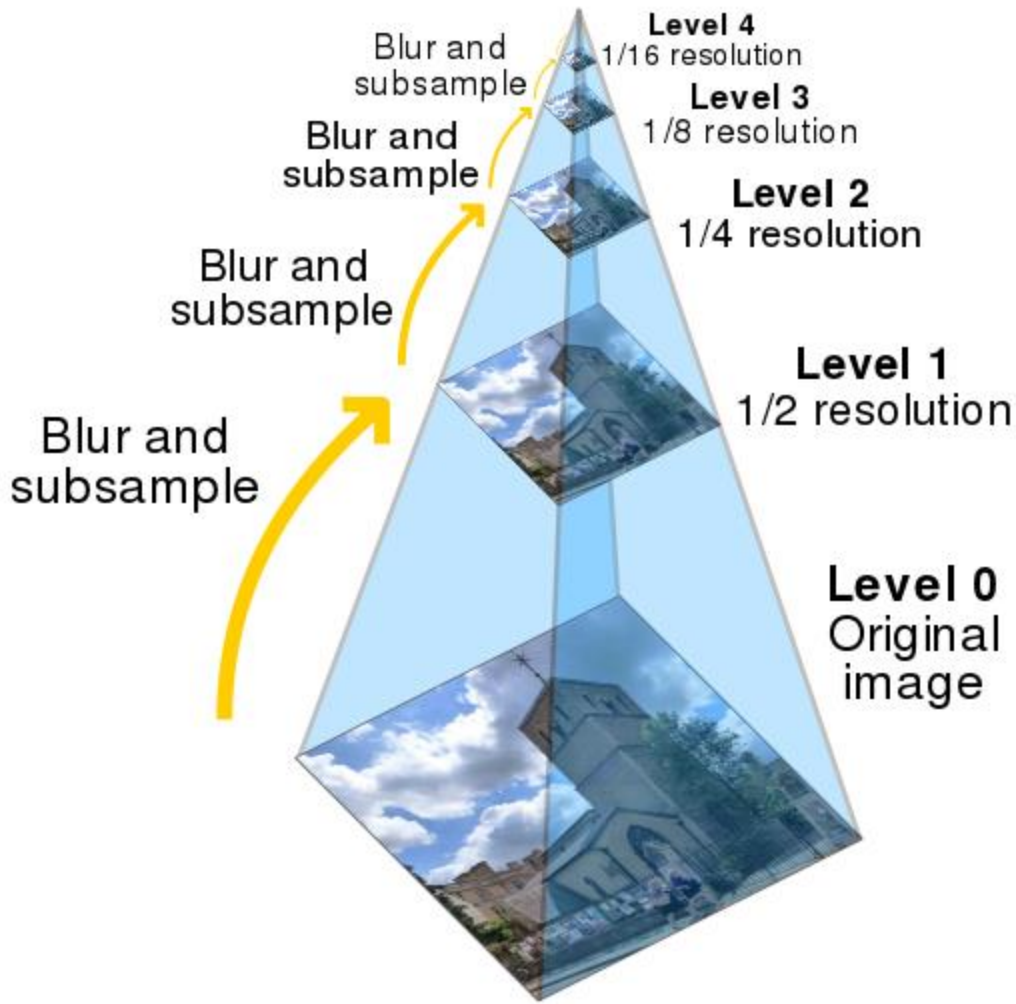
# Gaussian pre-filtering

- Solution: filter the image, *then* subsample

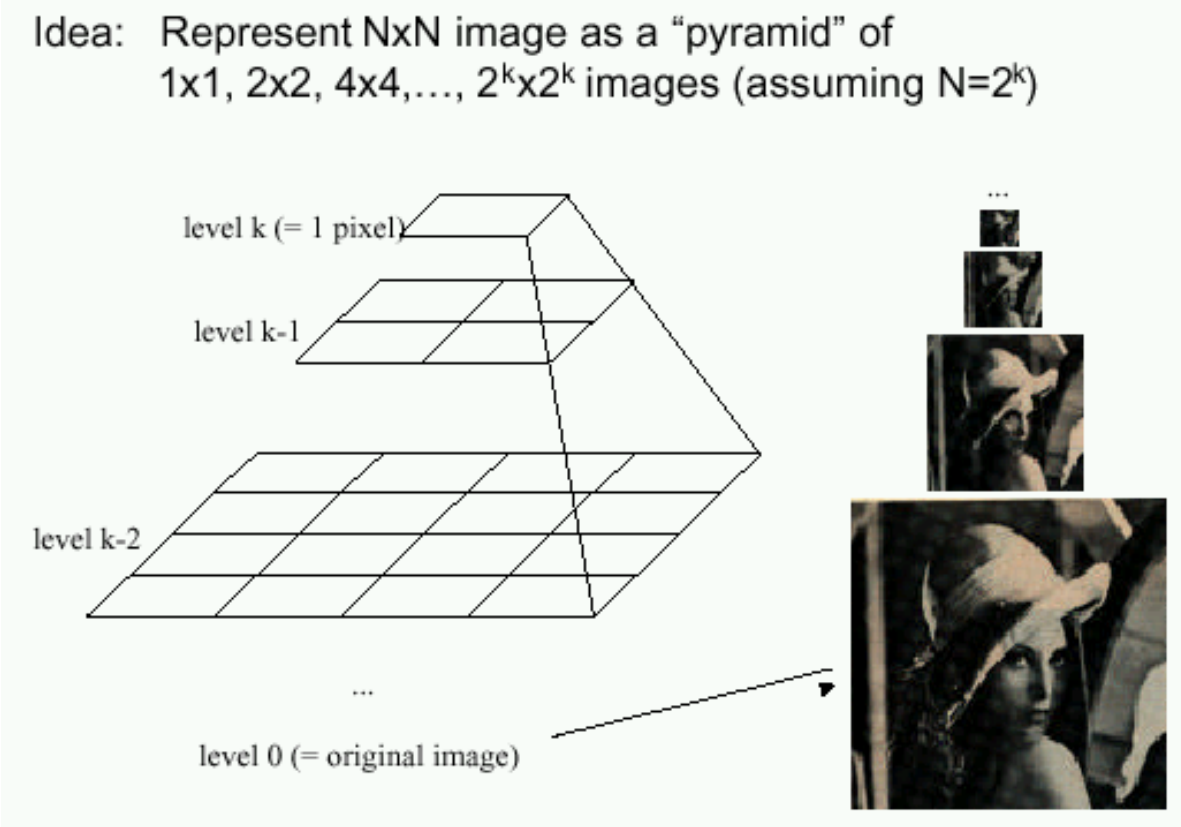


*Gaussian pyramid*




Level 4
1/16 resolution
Blur and subsample
Level 3
1/8 resolution
Blur and subsample
Level 2
1/4 resolution
Blur and subsample
Level 1
1/2 resolution
Blur and subsample
Level 0
Original image

# **Gaussian pyramids** [Burt and Adelson, 1983]

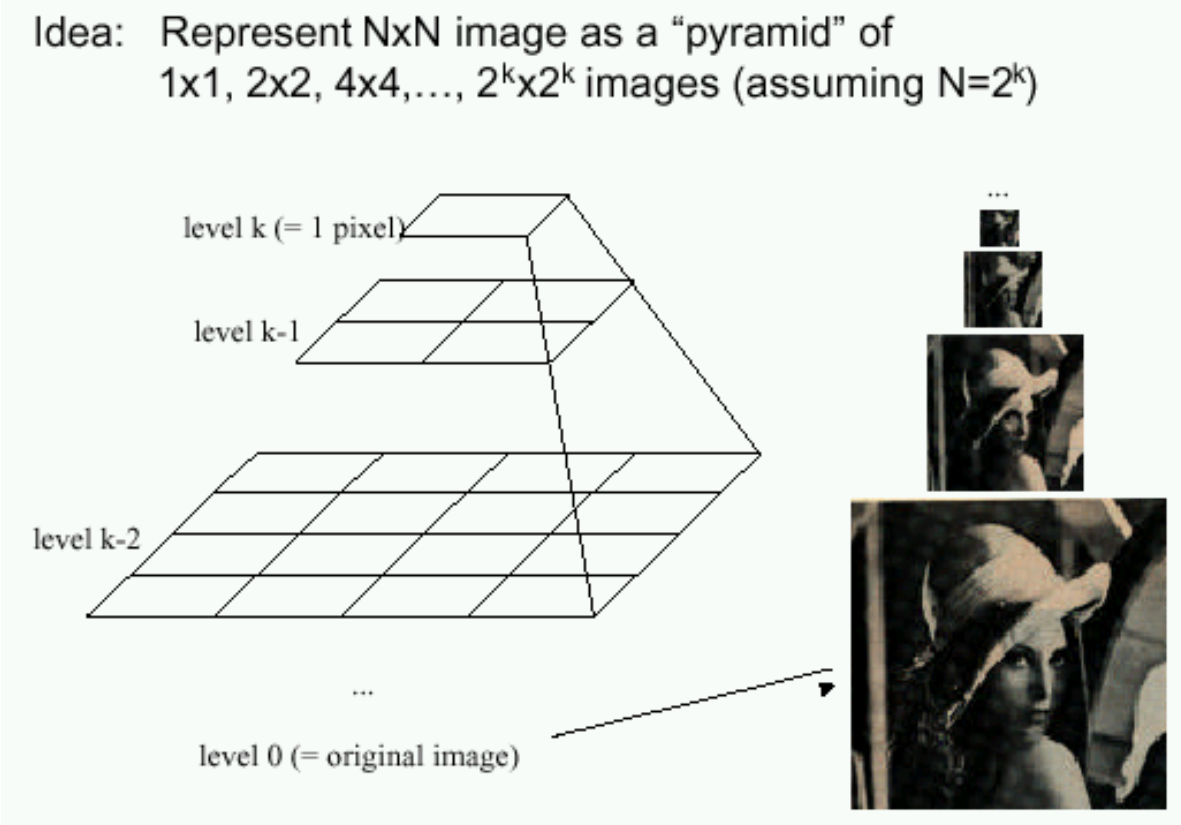



- In computer graphics, a *mip map* [Williams, 1983]
- A precursor to *wavelet transform*

Gaussian Pyramids have all sorts of applications in computer vision

Source: S. Seitz

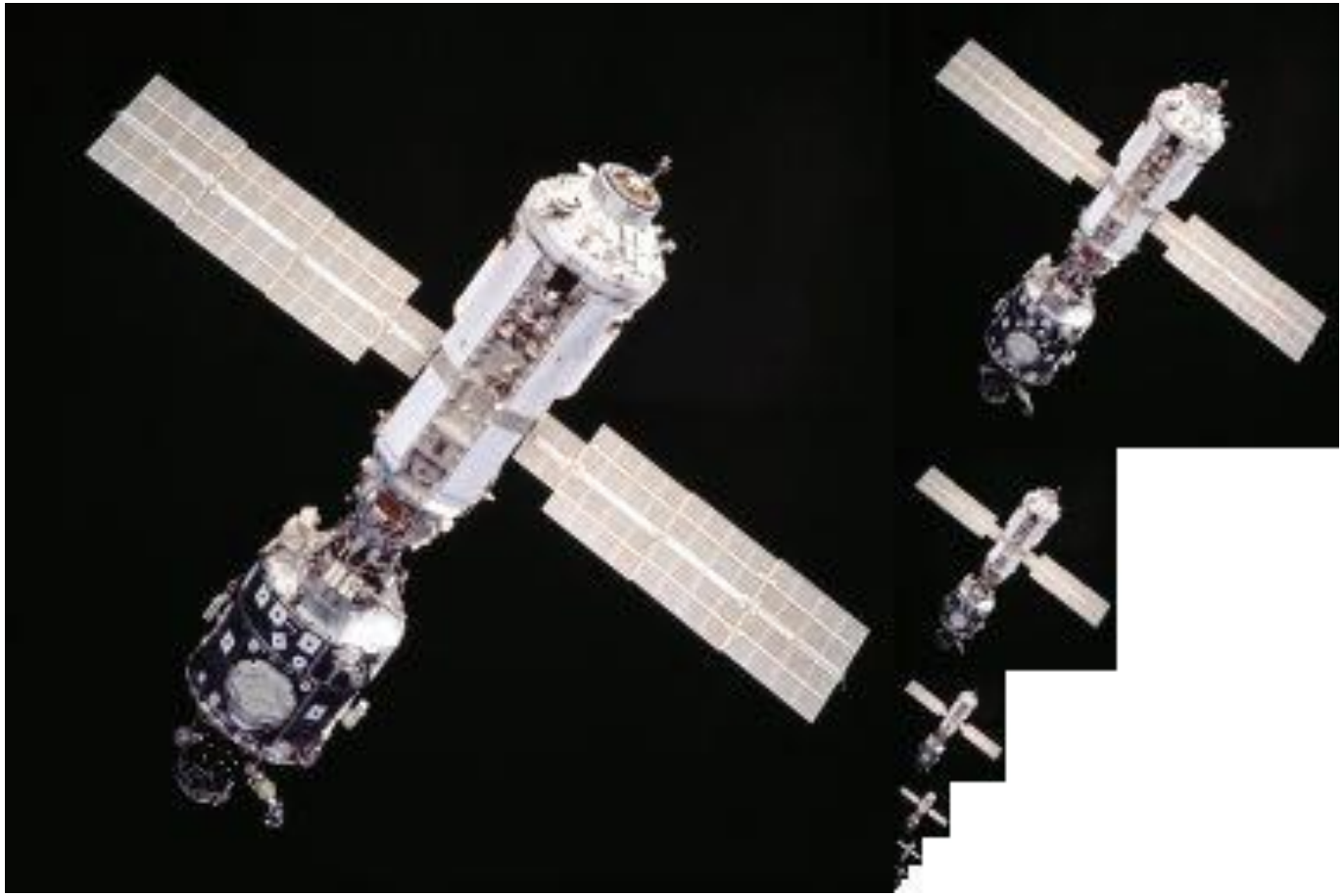
# **Gaussian pyramids** [Burt and Adelson, 1983]



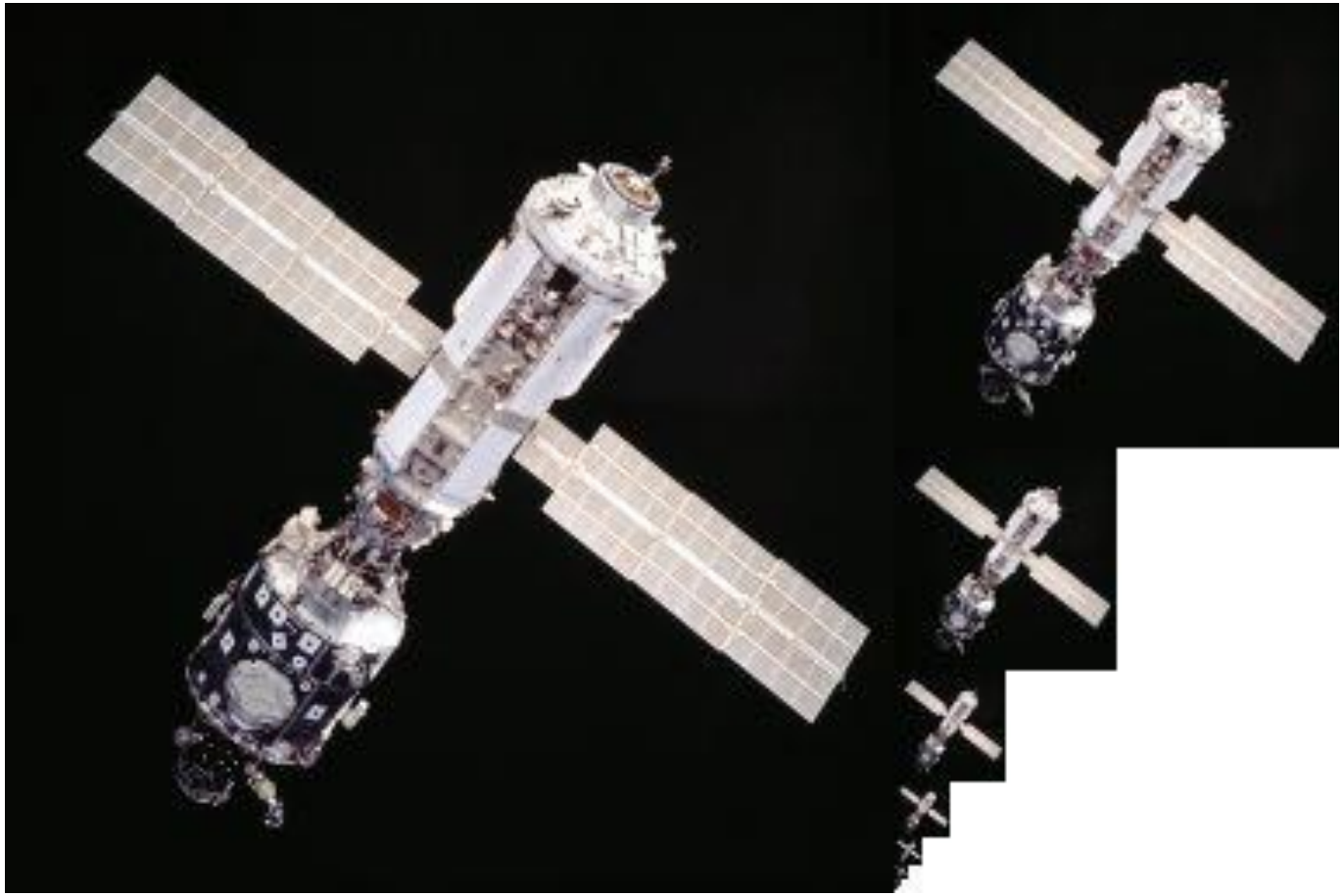

- How much space does a Gaussian pyramid take compared to the original image?

Source: S. Seitz

# Gaussian pyramid

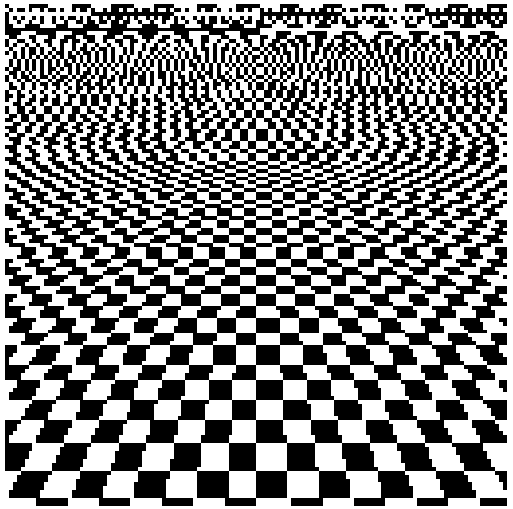


# Gaussian pyramid

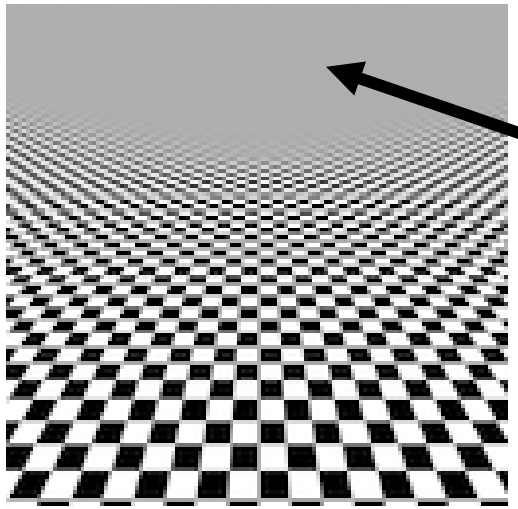


# Back to the checkerboard

- What should happen when you make the checkerboard smaller and smaller?



Naïve subsampling



Proper prefiltering
(“antialiasing”)

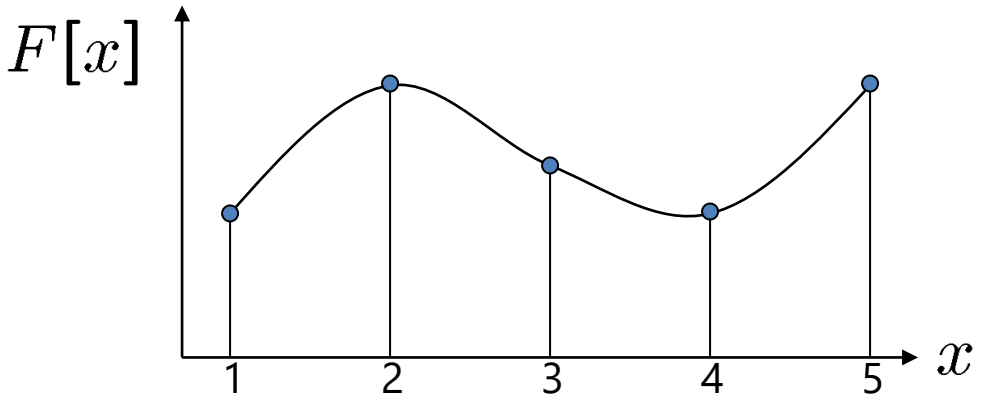
Image turns grey!
(Average of black and white squares, because each pixel contains both.)

# Questions?

# Upsampling

- This image is too small for this screen: 
- How can we make it 10 times as big?
- Simplest approach:
  repeat each row
  and column 10 times
- (“Nearest neighbor
  interpolation”)

# Image interpolation




d = 1 in this example

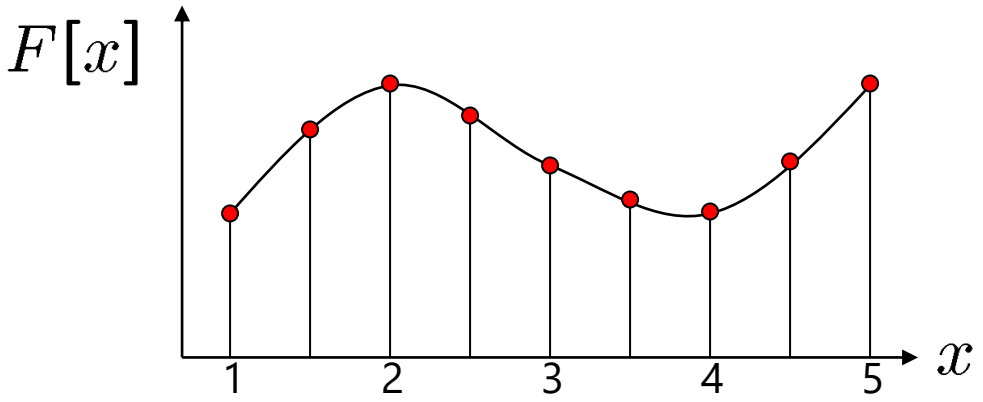
Recall that a digital images is formed as follows:

$$F[x, y] = \text{quantize}\{f(xd, yd)\}$$

- It is a discrete point-sampling of a continuous function
- If we could somehow reconstruct the original function, any new image could be generated, at any resolution and scale

Adapted from: S. Seitz

# Image interpolation




d = 1 in this example

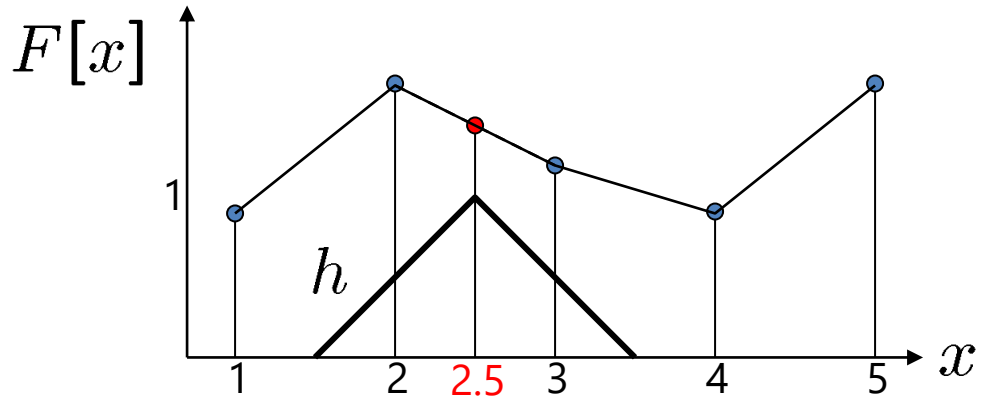
Recall that a digital images is formed as follows:

$$F[x, y] = \text{quantize}\{f(xd, yd)\}$$

- It is a discrete point-sampling of a continuous function
- If we could somehow reconstruct the original function, any new image could be generated, at any resolution and scale

Adapted from: S. Seitz

# Image interpolation




d = 1 in this example

- What if we don't know $f$ ?
  - Guess an approximation: $\tilde{f}$
  - Can be done in a principled way: filtering
  - Convert $F$ to a continuous function:

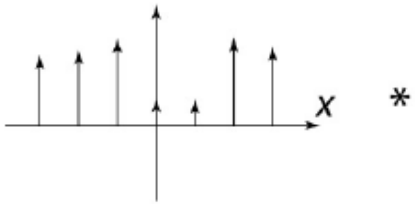
    $f_F(x) = F(\frac{x}{d})$ when $\frac{x}{d}$ is an integer, 0 otherwise

  - Reconstruct by convolution with a *reconstruction filter, h*

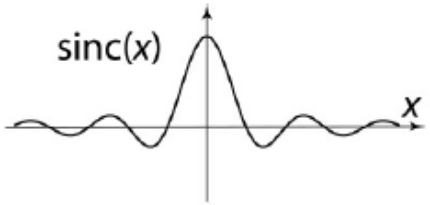
$$\tilde{f} = h * f_F$$

Adapted from: S. Seitz

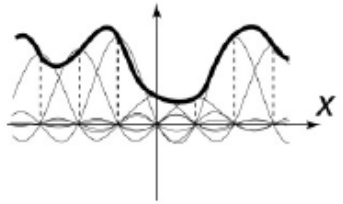
# Image interpolation



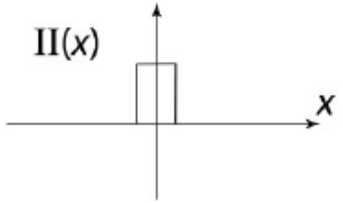

*



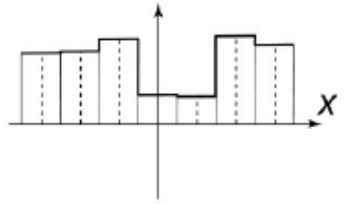

⇒



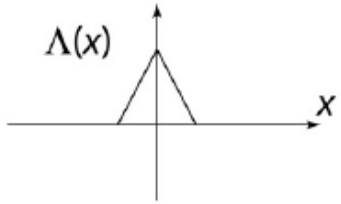

"Ideal" reconstruction



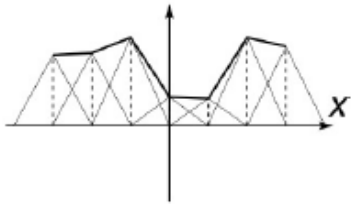

⇒



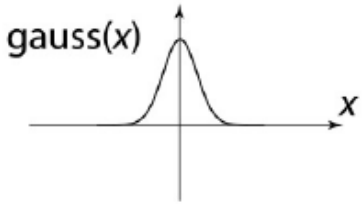

Nearest-neighbor interpolation



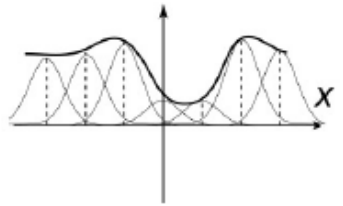

⇒




Linear interpolation




⇒



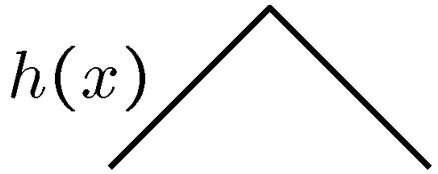

Gaussian reconstruction

Source: B. Curless

# Reconstruction filters

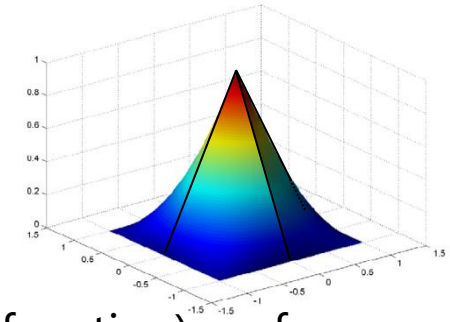
- What does the 2D version of this hat function look like?

$h(x)$



performs
linear interpolation

$h(x, y)$



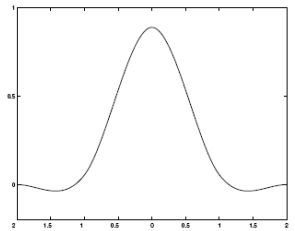
(tent function) performs
**bilinear interpolation**

Often implemented without cross-correlation

- E.g., http://en.wikipedia.org/wiki/Bilinear_interpolation

Better filters give better resampled images

- **Bicubic** is common choice



Cubic reconstruction filter

$$r(x) = \frac{1}{6}\begin{cases} (12-9B-6C)|x|^3 + (-18+12B+6C)|x|^2 + (6-2B) & |x|<1 \\ ((-B-6C)|x|^3 + (6B+30C)|x|^2 + (-12B-48C)|x| + (8B+24C) & 1\le|x|<2 \\ 0 & \textit{otherwise} \end{cases}$$

# Image interpolation

Original image:  x 10

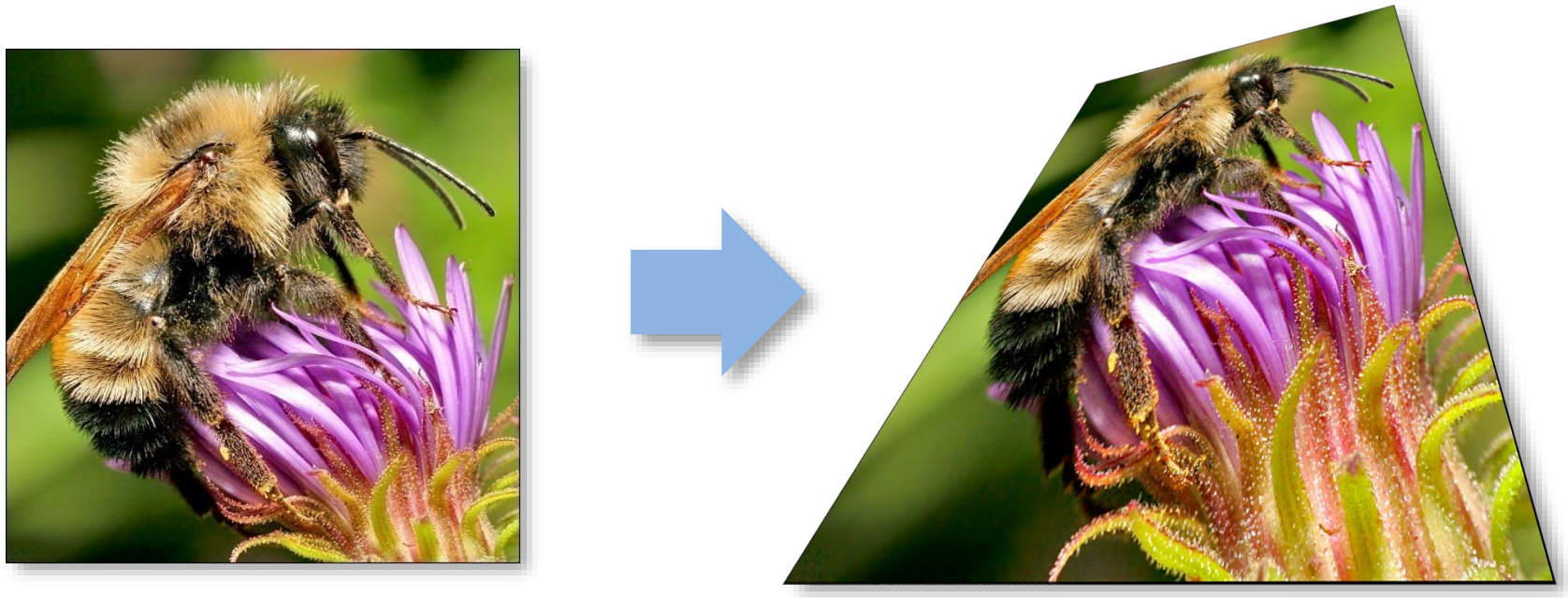
Nearest-neighbor interpolation

Bilinear interpolation

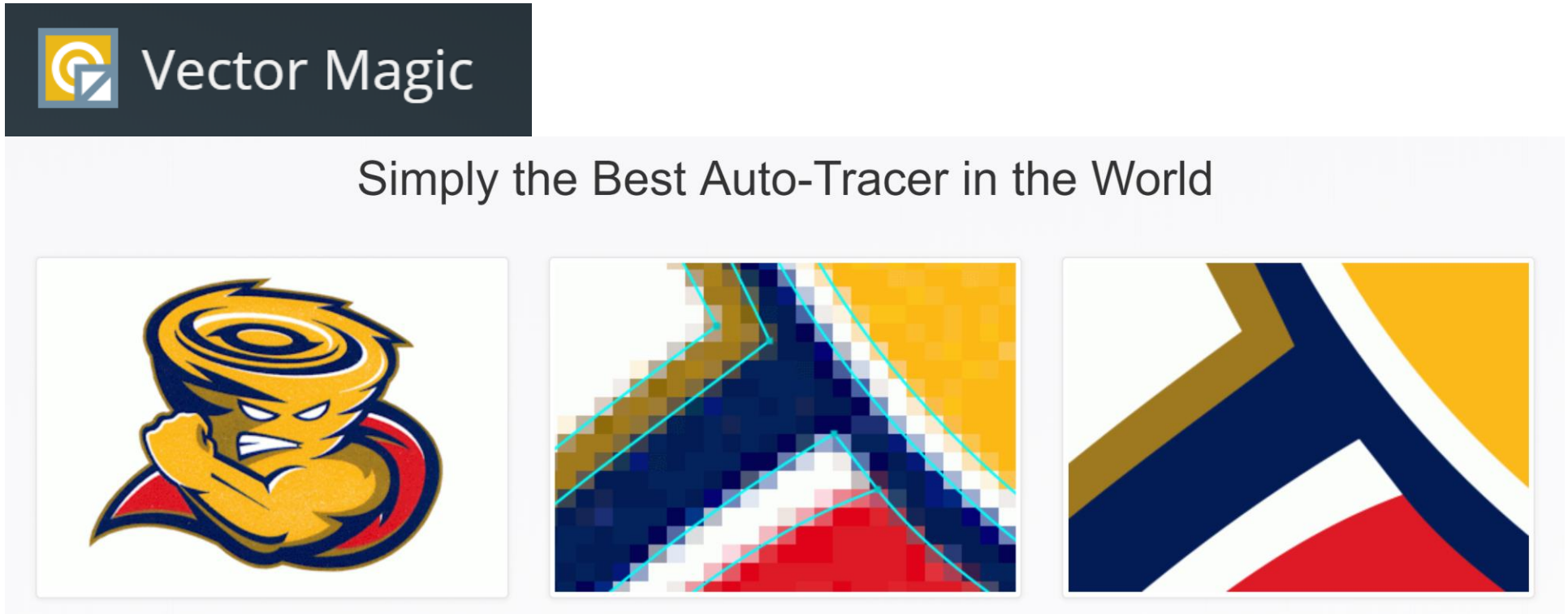
Bicubic interpolation

# Image interpolation

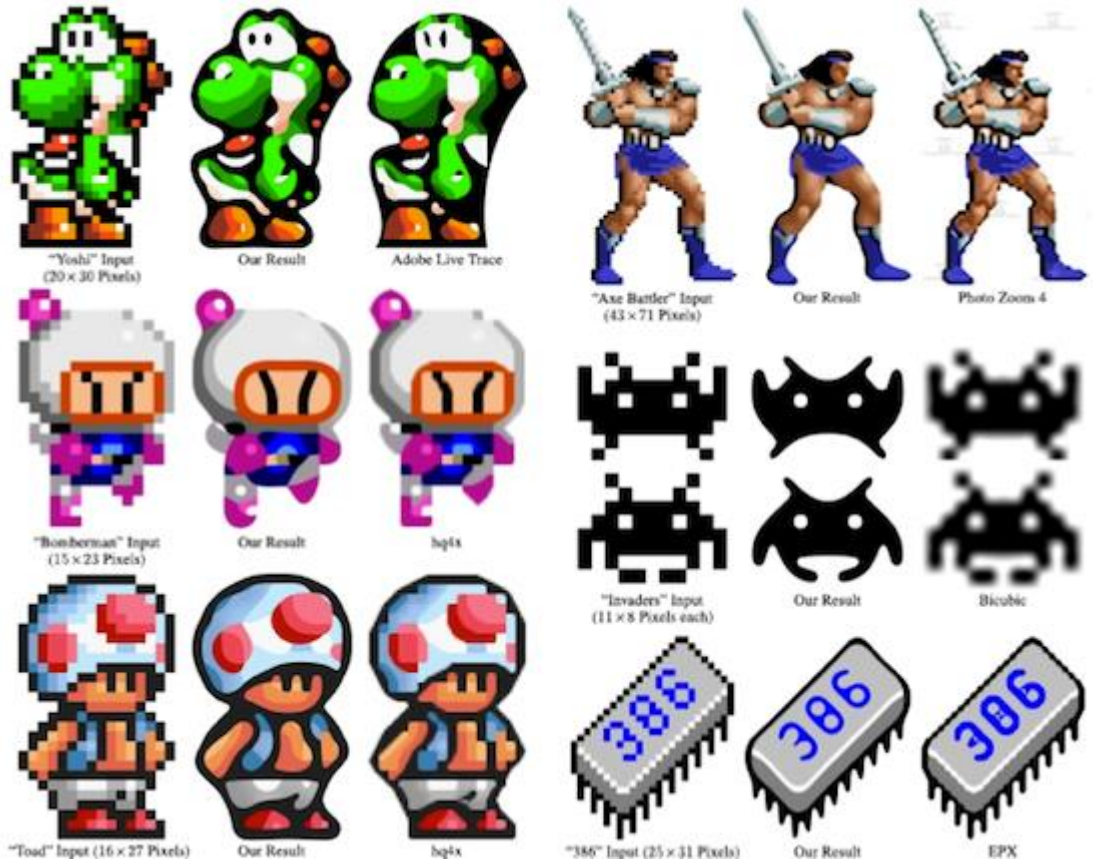
Also used for *resampling*



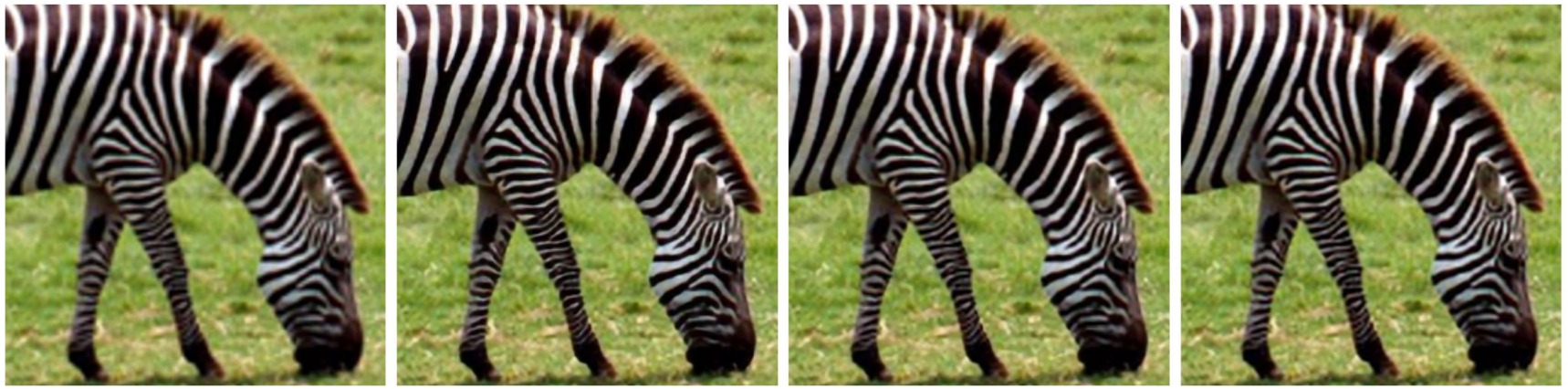
# Raster-to-vector graphics



# Depixelating Pixel Art



# Modern methods



(a) Bicubic (b) SRCNN (c) A+ (d) RAISR

(e) Bicubic (f) SRCNN (g) A+ (h) RAISR

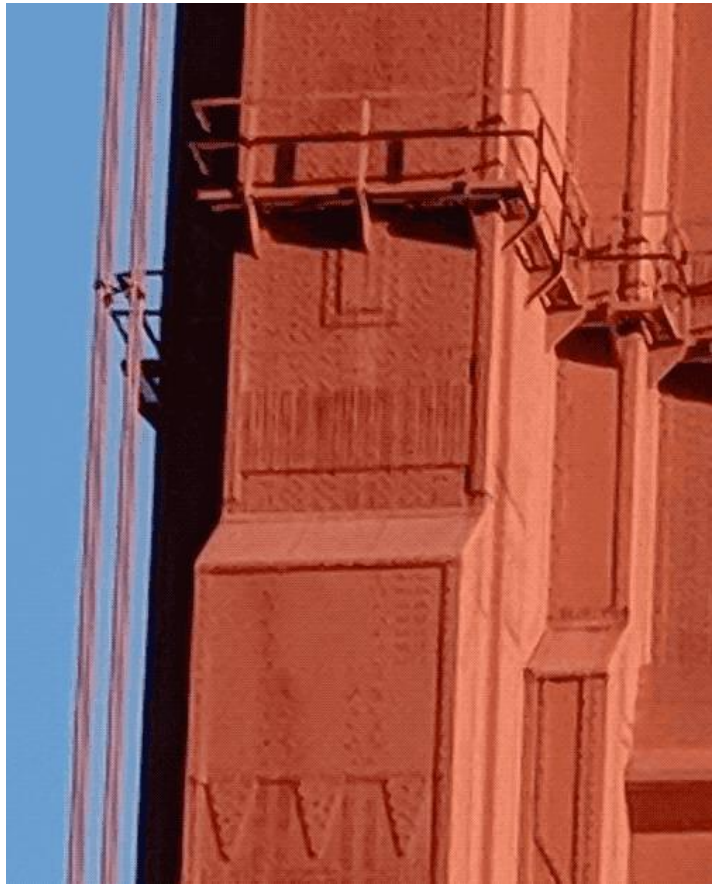
From Romano, et al: RAISR: Rapid and Accurate Image Super Resolution, https://arxiv.org/abs/1606.01299

# Super-resolution with multiple images

- Can do better upsampling if you have multiple images of the scene taken with small (subpixel) shifts
- Some cellphone cameras (like the Google Pixel line) capture a **burst** of photos
- Can we use that burst for upsampling?

# Google Pixel 3 Super Res Zoom

Effect of hand tremor as seen in a cropped burst of photos, after global alignment



Example photo with and without super res zoom (smart burst align and merge)

https://ai.googleblog.com/2018/10/see-better-and-further-with-super-res.html

# Questions?